Marginal Boosting[1]

Gunnar Rätsch[2] Manfred K. Warmuth[3]

NeuroCOLT2 Technical Report Series

NC2-TR-2001-097

August 7, 2001

Produced as part of the ESPRIT Working Group in Neural and Computational Learning II, NeuroCOLT2 27150

[1]Supported by DFG grant MU 987/1-1, EU-Neurocolt II and NSF grant CCR 9821087.
[2]raetsch@first.gmd.de GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany
[3]manfred@cse.ucsc.edu University of California at Santa Cruz, CA 95060, USA

Abstract

AdaBoost produces a linear combination of weak hypotheses. It has been observed in practice that the generalization error of the algorithm continues to improve even after all examples are classified correctly by the current linear combination, i.e. by a *hyperplane* in feature space where each weak hypothesis is a dimension. The improvement is attributed to the experimental observation that the distances (margins) of the examples to the separating hyperplane are increasing even when the training error is already zero, i.e. all examples are on the correct side of the hyperplane. We give an iterative version of AdaBoost that explicitly maximizes the minimum margin of the examples. We bound the number of iterations and the number of hypotheses used in the final linear combination which approximates the maximum margin hyperplane with a certain precision. This result is shown to be independent from the size of the hypothesis class – even infinite hypothesis classes are allowed.

1 Introduction

In the most common version of boosting the algorithm is given a fixed set of labeled training examples. In each stage the algorithm produces a probability weighting on the examples. It then is given a weak hypothesis whose error (probability of wrong classification) is slightly below 50%, which is used to update the distribution. Intuitively, the hard examples receive high weights. At end of each stage the weak hypothesis is added to the linear combination, which forms the current hypothesis of the boosting algorithm.

The most well known boosting algorithm is AdaBoost [4]. It adapts the linear coefficient of the weak hypothesis to the error of the weak hypothesis. Earlier work on boosting includes [12, 2]. AdaBoost has two redeeming properties. First, along with earlier boosting algorithms [12], it has the property that its training error converges exponentially fast to zero. More precisely, if the training error of the t-th weak learner is $\hat{\epsilon}_t = \frac{1}{2} - \frac{1}{2}\hat{\gamma}_t$, then an upper bound on the training error of the linear combination is reduced by a factor of $1 - \hat{\gamma}_t^2$ at stage t. Second, it has been observed experimentally that AdaBoost continues to "learn" even after the training error of the linear combination is zero [13], i.e. in experiments the generalization error is continuing to improve. When the training error is zero, then all examples are on the "right side" of the linear combination (viewed as a *hyperplane* in a feature space, where each base hypothesis is one dimension). The *margin* of an example is the signed distance to the hyperplane times its $\pm$ label. As soon as the training error is zero, the examples are on the right side and have positive margin. It has also been observed that the margins of the examples continue to increase even after the training error is zero. There are theoretical bounds on the generalization error of linear classifiers (e.g. [13, 1]) that improve with the size of the minimum margin of the examples. So the fact that the margins improve experimentally seems to explain why AdaBoost still learns after the training error is zero.

There is one shortfall in this argument. AdaBoost has not been proven to maximize the minimum margin of the examples. In fact, in our experiments in Section 4 we observe that AdaBoost does not seem to maximize the margin. Breiman [1] proposed a modified algorithm – Arc-GV (**Arc**ing-**G**ame **V**alue) – suitable for this task and

showed that it *asymptotically* maximizes the margin. In this paper we propose an algorithm that maximizes the margin up to a given accuracy ε. We prove *exponential convergence rates* to the maximum margin solution in terms of ε and the sample size N. To our knowlegde, this is the first result on the *non-asymptotical* convergence of a boosting algorithm to the maximum margin solution.

The paper is structured as follows: In Section 2 we first extend the original AdaBoost algorithm leading to *AdaBoost*$_\varrho$. Then we propose *Marginal AdaBoost*, which uses AdaBoost$_\varrho$ as a subroutine. In Section 3 we give a more detailed analysis of both algorithms. First, we prove that if the training error of the t-th weak learner is $\hat{\epsilon}_t = \frac{1}{2} - \frac{1}{2}\hat{\gamma}_t$, then an upper bound on the fraction of examples with margin smaller than ϱ is reduced by a factor of $1 - (\varrho - \hat{\gamma}_t)^2$ at stage t of AdaBoost$_\varrho$ (cf. Theorem 2 and Corollary 3). Exploiting this property, we prove the exponential convergence rate of our algorithm (cf. Theorem 4). We complete the paper with experiments confirming our theoretical analysis (Section 4) and a conclusion.

2 Marginal Boosting

For AdaBoost it has been shown that it quickly generates a combined hypothesis $f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_t \frac{\alpha_t}{\sum_r \alpha_r} h_t(\mathbf{x})$ consistent with the training set [4]. This is desirable for the analysis in the PAC setting [12, 15]. Consistency on the training set means that the margin of each training example is larger than zero, i.e. $\min_{n=1,\dots,N} y_n f(\mathbf{x}_n) > 0$. We start with a slight modification of AdaBoost (cf. Algorithm 1), which does not only aim to find a hypothesis with margin of at least zero, but with at least margin ϱ, where ϱ is pre-specified. We call this algorithm AdaBoost$_\varrho$, as it naturally generalizes

Algorithm 1 The AdaBoost$_\varrho$ algorithm.

1. **Input:** $S = \langle(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\rangle$, No. of Iterations T, margin target ϱ
2. **Initialize:** $d_n^1 = 1/N$ for all $i = 1 \dots N$
3. **Do for** $t = 1, \dots, T$,
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$
   (b) Calculate the edge $\hat{\gamma}_t$ of h_t: $\hat{\gamma}_t = \sum_{i=1}^{l} d_n^t y_n h_t(\mathbf{x}_n)$.
   (c) Set $\alpha_t = \frac{1}{2}\log\frac{1+\hat{\gamma}_t}{1-\hat{\gamma}_t} - \frac{1}{2}\log\frac{1+\varrho}{1-\varrho}$.
   (d) Update weights: $d_n^{t+1} = d_n^t \exp\{-\alpha_t y_n h_t(\mathbf{x}_n)\}/Z_t$, s.t. $\sum_{n=1}^{N} d_n^{t+1} = 1$.
4. **Break if** $\hat{\gamma}_t \leq \varrho$ or $\hat{\gamma}_t = 1$.
5. **Output:** $f(\mathbf{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_r \alpha_r} h_t(\mathbf{x})$, $\hat{\gamma} = \min_{t=1,\dots,T} \hat{\gamma}_t$, and $\hat{\varrho} = \max_{n=1,\dots,N} y_n f(\mathbf{x}_n)$

AdaBoost for the case when the *target margin* is ϱ. The original AdaBoost algorithm now becomes $AdaBoost_0$, as it targets $\varrho = 0$. The algorithm AdaBoost$_\varrho$ is already known as *unnormalized Arcing* [1] or *AdaBoost-type Algorithm* [11]. The only difference to AdaBoost is the choice of the hypothesis coefficients α_t. There appears an additional term, i.e. $-\frac{1}{2}\log\frac{1+\varrho}{1-\varrho}$, which is zero for AdaBoost$_0$. The constant ϱ might be seen as a *guess* of the maximum margin ϱ^*. If ϱ would be chosen properly (close to

ϱ^*), AdaBoost$_\varrho$ would converge fast to a combined hypothesis with a near maximum margin. The details are given in Section 3.2.

Since one does not know the value of ϱ^* beforehand, one also needs to find ϱ^*. We propose an algorithm that constructs a sequence $\{\varrho_r\}_{r=1}^R$ converging to ϱ^*: A fast way to find a real value up to a certain accuracy ε on the interval $[-1,1]$ is to use a *binary search* – one needs only $\log_2(2/\varepsilon)$ steps. Our idea is to use the binary search to find ϱ^*, where we call Algorithm 1 to decide whether the current guess ϱ is *larger* or *smaller* than ϱ^*. This leads to Algorithm 2.

The algorithm proceeds in R iterations, where R is determined by the accuracy ε that we would like to reach: In each iteration r it calls AdaBoost$_{\varrho_r}$ (cf. step 3a in Algorithm 2), where ϱ_r is chosen to be in the middle of an interval $[l_r, u_r]$ (cf. step 3c). Based on the success of AdaBoost$_{\varrho_r}$ to achieve a large enough margin, the interval is updated (cf. step 3b). We can show that the interval is chosen such that it always contains ϱ^*, the *unknown* maximal margin, while the length of the interval is almost reduced by a factor of two. Finally, in the last step of the algorithm, one has reached a good estimate ϱ_R of ϱ^* and calls AdaBoost$_\varrho$ for $\varrho = l_{R+1} - \varepsilon$ generating a combined hypothesis with margin at least $\varrho^* - 4\epsilon$.

In the next section we will give a detailed analysis of how the algorithm works and how many calls to the base learner are needed to approximate ϱ^*: in the worst case one needs about $\log_2(2/\varepsilon)$ times more iterations than in the case where we already know ϱ^*.

Since AdaBoost$_\varrho$ is used as a sub-routine and starts from the beginning in each round, one can think of several speed-ups, which might help to reduce the computation time. For instance, one could store the base hypotheses of previous iterations and instead of calling the base learner, one first sifts through these previously used hypotheses. Furthermore, one may stop AdaBoost$_\varrho$, when the combined hypothesis has reached a margin of ϱ or the base learner returns a hypothesis with edge lower than ϱ.

Algorithm 2 The Marginal AdaBoost algorithm

1. **Input:** $S = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\rangle$, Accuracy ε
2. **Initialize:** $\varrho_1 = 0$, $l_1 = -1$, $u_1 = 1$, $R = \lceil \log_2(1/\varepsilon) \rceil$, $T = \lceil 2\log(N)/\varepsilon^2 \rceil + 1$.
3. **Do for** $r = 1, \ldots R$,
   (a) $[f_r, \hat{\gamma}_r, \hat{\varrho}_r] = \mathrm{AdaBoost}_{\varrho_r}(S, T)$
   (b) **if** $\hat{\varrho}_r \geq \varrho_r$, **then** $l_{r+1} = \max(\hat{\varrho}_r, l_r)$, $u_{r+1} = \min(\hat{\gamma}_r, u_r)$
   else $l_{r+1} = \max(\hat{\varrho}_r, l_r)$, $u_{r+1} = \min(\hat{\gamma}_r, \varrho_r + \varepsilon, u_r)$
   (c) $\varrho_{r+1} = \frac{l_r + u_r}{2}$
4. **Break if** $u_{r+1} - l_{r+1} \leq 3\varepsilon$
5. **Output:** $f = \mathrm{AdaBoost}_{l_{r+1}-\varepsilon}(S, 2\log(N)/\varepsilon^2)$

3 Detailed Analysis

3.1 Weak learning and margins

The standard assumption made on the base learning algorithm in the PAC-Boosting setting is that it returns a hypothesis h from a fixed set H that is slightly better than random guessing on any training set.[1] More formally this means that the error rate $\hat{\epsilon}$ is consistently smaller than $\frac{1}{2} - \frac{1}{2}\hat{\gamma}$. Note that the error rate of $\frac{1}{2}$ (i.e. $\hat{\gamma} = 0$) could easily be reached by a fair coin, assuming both classes have the same prior probabilities. Thus, this requirement is the least we can expect from a learning algorithm [4].

The error rate $\hat{\epsilon}$ is defined as the fraction of points that are misclassified. In Boosting this is extended to weighted sample sets and one defines

$$\hat{\epsilon}(\mathbf{d}, h) = \sum_{n=1}^{N} d_n \, \mathbf{I}(y_n \neq \text{sign}(h(\mathbf{x}_n))),$$

where h is the hypothesis returned by the base learner and $\mathbf{I}$ is the identity function with $\mathbf{I}(\text{true}) = 1$ and $\mathbf{I}(\text{false}) = 0$. The weighting $\mathbf{d} = [d_1, \ldots, d_N]$ of the examples is such that $d_n \geq 0$ and $\sum_{n=1}^{N} d_n = 1$. A more convenient quantity to measure the quality of the hypothesis h is the edge $\hat{\gamma}(\mathbf{d}, h) = \sum_{n=1}^{N} d_n y_n h(\mathbf{x}_n)$ [1], which is an affine transformation of $\hat{\epsilon}(\mathbf{d}, h)$, if $h(\mathbf{x}) \in \{-1, +1\}$: $\hat{\epsilon}(\mathbf{d}, h) = \frac{1}{2} - \frac{1}{2}\hat{\gamma}(\mathbf{d}, h)$. Recall from Section 1 the definition of the margin $\hat{\rho}(\boldsymbol{\alpha}, \mathbf{z}) = y f_{\boldsymbol{\alpha}}(\mathbf{x})$ of a given example $\mathbf{z} = (\mathbf{x}, y)$.

Suppose we would combine all possible hypotheses from H (assuming H is finite), then the following well-known theorem establishes the connection between margins and edges :

Theorem 1 (Min-Max-Theorem [16], see also [3, 1]).

$$\gamma^* = \min_{\mathbf{d}} \max_{h \in H} \hat{\gamma}(\mathbf{d}, h) = \max_{\boldsymbol{\alpha}} \min_{n=1,\ldots,N} \hat{\rho}(\boldsymbol{\alpha}, \mathbf{z}_n) = \varrho^*, \tag{1}$$

where $\mathbf{d} \in \mathcal{P}^N$, $\boldsymbol{\alpha} \in \mathcal{P}^{|H|}$ *and* $\mathcal{P}^k$ *is the* k-dimensional probability simplex.

Thus, the minimal edge $\hat{\gamma}$ that can be achieved over all possible weightings $\mathbf{d}$ of the training set is equal to the maximal margin of a combined hypothesis from H. Also, for any non-optimal weightings $\mathbf{d}$ and $\boldsymbol{\alpha}$ we always have $\max_{h \in H} \hat{\gamma}(\mathbf{d}, h) > \min_{n=1,\ldots,N} \hat{\rho}(\boldsymbol{\alpha}, \mathbf{z}_n)$. So, assuming that the base learning algorithm performs optimal, i.e. maximizes the edge for a given $\mathbf{d}$, one can conclude $\hat{\gamma} \geq \hat{\varrho}$ (cf. Algorithm 1). If the base learning algorithm guarantees to return a hypothesis with edge at least $\hat{\gamma}$ for any weighting, there exists a combined hypothesis with margin at least $\hat{\gamma}$. If $\hat{\gamma} = \gamma^*$, i.e. the lower bound $\hat{\gamma}$ is largest possible, then there exists a combined hypothesis with margin exactly $\hat{\gamma} = \varrho^*$. From this discussion we can derive a sufficient condition on the base learning algorithm to reach the maximal margin: If it returns hypotheses with edge at least γ^*, one is able to combine these hypotheses with margin ϱ^*. This explains the termination condition (cf. step 4 in Algorithm 1).

[1] In the PAC setting, the base learner is allowed to fail with probability δ. Since we are seeking for simple presentation, we ignore this fact here. Our algorithm can be extended to this case.

Note, for AdaBoost$_0$ it sofar has only been shown that it *asymptotically* achieves a margin of at least $\varrho^*/2$ [11]. We aim to find an algorithm that approximates the maximum margin solution up to any precision in *few iterations.*

3.2 Convergence properties of AdaBoost$_\varrho$

We now analyze a slightly generalized version of Algorithm 1, where ϱ is not fixed but could be adapted in each iteration. We therefore consider sequences $\{\varrho_t\}_{t=1}^T$, which might either be specified before running the algorithm or computed based on results during the algorithm. For instance, the idea proposed by [1] is to set $\varrho_t = \hat{\varrho}_{t-1} := \min_{n=1,\dots,N} \hat{\rho}(\boldsymbol{\alpha}_{t-1}, \mathbf{z}_n)$, which leads to Arc-GV. We are answering the question how good AdaBoost$_{\{\varrho_t\}}$ is able to increase the margin and bound the fraction of examples, which have a margin smaller than say θ. This leads to a theorem generalizing Thm. 5 in [4] for the case $\varrho \neq 0$:

Theorem 2 ([11, 9]). *Let $\hat{\gamma}_1, \dots, \hat{\gamma}_T$ be the edges of $h_1, \dots, h_T$ that are generated by Algorithm 1 and $-1 \leq \varrho_t \leq \hat{\gamma}_t$ for $t = 1, \dots, T$. Then for all $\theta \in [-1, 1]$*

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}\left(y_n f(\mathbf{x}_n) \leq \theta\right) \leq \prod_{t=1}^{T} \sqrt{\left(\frac{1-\hat{\gamma}_t}{1-\varrho_t}\right)^{1-\theta} \left(\frac{1+\hat{\gamma}_t}{1+\varrho_t}\right)^{1+\theta}}, \tag{2}$$

where f is the final hypothesis. Furthermore, for $\theta \leq \min_{t=1,\dots,T} \varrho_t$,

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}\left(y_n f(\mathbf{x}_n) \leq \theta\right) \leq \exp\left\{-\sum_{t=1}^{T} \Delta_2(\varrho_t, \hat{\gamma}_t)\right\}, \tag{3}$$

where $\Delta_2(\varrho, \hat{\gamma}) := \frac{1+\varrho}{2}\log\frac{1+\varrho}{1+\hat{\gamma}} + \frac{1-\varrho}{2}\log\frac{1-\varrho}{1-\hat{\gamma}}$ is the binary relative entropy.

Thus, the algorithm makes progress reducing the rhs. of (2), if the term under the square-root is smaller then one. This is e.g. the case if $\hat{\gamma}_t$ large compared to θ and ϱ_t or, by (3), if $\theta \leq \varrho_t < \hat{\gamma}_t$ (cf. step 4 in Algorithm 1). The larger $\hat{\gamma}_t$, the more progress one makes.

Suppose we would like the reach a margin θ on all training examples, where we obviously need to assume $\theta \leq \varrho^*$. Then the question arises, which sequence of $\{\varrho_t\}_{t=1}^T$ one should use to find a combined hypothesis in as few iterations as possible. One can bound $\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}(y_n f(\mathbf{x}_n) \leq \theta) \leq (2) = \prod_{t=1}^{T} \exp\{\theta\alpha_t + \log \hat{Z}_t\}$, where α_t is given in step (3c) and $\hat{Z}_t = \frac{1+\varrho}{2}\exp(-\alpha_t) + \frac{1-\varrho}{2}\exp(\alpha_t)$ is a upper bound [14] on Z_t as used in step (3d) of Algorithm 1. We achieve a minimum of the left hand side, if we independently chose ϱ_t in each iteration t such that $\exp\{\theta\alpha_t + \log \hat{Z}_t\}$ is minimized. Setting the derivative with respect to ϱ_t to zero and solving for ϱ_t yields $\varrho_t = \theta$. *So it turns out that one should use $\varrho_t \equiv \theta$, independently how the base learner performs!* This explains why we use $\varrho = \text{const}$.

The number of iterations needed to achieve a margin of at least ϱ can be upper bounded:

Corollary 3. *Assume the base learner always achieves an edge $\hat{\gamma}_t \geq \varrho^*$. If $\varrho \leq \varrho^* - \varepsilon$, $\varepsilon > 0$, then AdaBoost$_\varrho$ will converge to a solution with margin of at least ϱ on all examples in at most $\lceil \frac{2\log(N)}{\varepsilon^2} \rceil + 1$ steps.*

Proof. We use (3) for $\theta \equiv \varrho$, yielding (3) $\leq \exp\{-\sum_t \frac{1}{2}(\varrho - \hat{\gamma}_t)^2\} \leq \exp\{-\frac{T\varepsilon^2}{2}\}$, where we use a bound on the binary entropy. If $\exp\{-\frac{T\varepsilon^2}{2}\} < \frac{1}{N}$, there is no example left with margin smaller than ϱ, which proves the corollary. □

3.3 Convergence of Marginal AdaBoost

So far we have always considered the case where we already know some proper value of ϱ. Let us now assume the case that the maximum achievable margin is ϱ^* and we would like to achieve a margin of $\varrho^* - \varepsilon$. Our algorithm *guesses* a value of ϱ^* starting with 0. We need to understand what happens if our guess is too high, i.e. $\varrho > \varrho^*$, or too low, i.e. $\varrho < \varrho^*$?

First, if $\varrho > \varrho^*$, then one cannot reach the margin of ϱ since the maximum achievable margin is ϱ^*. By Corollary 3, if AdaBoost$_\varrho$ has not reached a margin of at least ϱ in $2\log(N)/\varepsilon^2$ steps, we can conclude that $\varrho > \varrho^* - \varepsilon$ (cf. step (3b) in Algorithm 2). This is the worst case. The better case is, if the distribution $\mathbf{d}$ generated by AdaBoost$_\varrho$ will be too difficult for the base learner and it eventually fails to achieve an edge $\hat{\gamma}$ of at least ϱ and AdaBoost$_\varrho$ will stop (cf. stopping condition in Algorithm 1).

Second, assume ϱ is chosen to low, say $\varrho < \varrho^* - \varepsilon$, then one achieves a margin of ϱ in a few steps by Corollary 3 . Since the maximum margin is always greater that a certain achieved margin, one can conclude that $\varrho^* \geq \varrho$ (cf. step (3b) in Algorithm 2).

Note that there is a *small gap* in the proposed binary search procedure: We are not able to identify the case $\varrho^* - \varepsilon \leq \varrho \leq \varrho^*$ efficiently. This means that we cannot reduce the length of the search interval by *exactly* a factor of two in *each* iteration. This makes the analysis slightly more difficult, but eventually leads to the following theorem on the *worst case performance* of Marginal AdaBoost:

Theorem 4. *Assume the base learner always achieves an edge $\hat{\gamma}_t \geq \varrho^*$. Then Algorithm 2 will find a combined hypothesis f that maximizes the margin up to accuracy 4ε in at most $\lceil \frac{2\log(N)}{\varepsilon^2} + 1 \rceil \lceil \log_2(1/\varepsilon) + 1 \rceil$ calls of the base learner. The final hypothesis combines at most $\lceil \frac{2\log(N)}{\varepsilon^2} + 1 \rceil$ base hypotheses.*

Proof. See Algorithm 2 for definitions of $u_r, l_r, \hat{\gamma}_r, \hat{\varrho}_r$.

We claim that in any iteration $u_r \geq \varrho^* \geq l_r$. We show, if $u_{r-1} \geq \varrho^* \geq l_{r-1}$, then $u_r \geq \varrho^* \geq l_r$ for all $r = 1, \ldots, R$. It holds $l_1 \geq \varrho^* \geq u_1$ (induction start). By assumption $\hat{\gamma}_{t,r} \geq \varrho^*$ and we may set $u_{r+1} = \min_{r_0=1,\ldots,r} \min_{t=1,\ldots,T} \hat{\gamma}_{t,r}$. By Theorem 1 holds $\varrho^* \geq \hat{\varrho}_r$ for all $r = 1, 2, \ldots$ and, hence, we may set $l_{r+1} = \max_{r_0=1,\ldots,r} \hat{\varrho}_{r_0}$. We have to consider two cases. (a) $\hat{\varrho}_r \geq \varrho_r$ and (b) $\hat{\varrho}_r < \varrho_r$. In case (b) we have an additional term in u_{r+1}, which follows from $\varrho_r + \varepsilon \geq \varrho^*$, justified by $\hat{\varrho}_r < \varrho_r$ and Theorem 2.

By construction, the length of interval $[l_r, u_r]$ is (almost) decreased in each iteration by a factor of two. We show $u_r - l_r \leq 2^{-r+1} + 2\varepsilon$. In case (a) the interval is reduced by at least a factor of two. The worst case is if always (b) happens:

$$\begin{aligned} u_r - l_r &\leq \varrho_r + \varepsilon - l_r \\ &= \frac{\varrho_{r-1} + \varepsilon + l_{r-1}}{2} + \varepsilon - l_r \\ &= \frac{\varrho_{r-1} + \varepsilon + l_{r-1} + 2\varepsilon - 2l_r}{2} \end{aligned}$$

$$
\begin{aligned}
&= \frac{\varrho_{r-j} + \varepsilon \sum_{i=0}^{j} 2^j + \sum_{i=1}^{j} 2^{j-i} l_{r-i} - 2^j l_r}{2^j} \\
&= \frac{\varrho_{r-j} + (2^{j+1}-1)\varepsilon + \sum_{i=1}^{j} 2^{j-i} l_{r-i} - 2^j l_r}{2^j} \\
&= \frac{\varrho_1 + (2^r-1)\varepsilon + \sum_{i=1}^{r-1} 2^{r-1-i} l_{r-i} - 2^{r-1} l_r}{2^{r-1}}.
\end{aligned}
$$

Since l_r is non-decreasing, $\sum_{i=1}^{j} 2^{r-1-i} l_{r-i} - 2^{r-1} l_r$ is maximized for $l_r = \text{const}$, i.e. $\sum_{i=1}^{j} 2^{r-1-i} l_{r-i} - 2^{r-1} l_r \leq l_r \sum_{i=0}^{r-2} 2^i - 2^{r-1} l_r = l_r(2^{r-1} - 1 - 2^{r-1}) = -l_r$. We continue by using $\varrho_1 = 0$ and $\min_r l_r = -1$:

$$
\begin{aligned}
&\leq \frac{(2^r-1)\varepsilon - l_r}{2^{r-1}} \\
&\leq \frac{1}{2^{r-1}} + 2\varepsilon.
\end{aligned}
$$

Thus after $R = \lceil \log_2(1/\varepsilon) \rceil$ steps we have $u_{R+1} - l_{R+1} \leq 3\varepsilon$ and $\varrho^* - l_{R+1} \leq 3\varepsilon$.

Now we run AdaBoost $_{l_{R+1}-\varepsilon}(S, 2\log(N)/\varepsilon^2)$ and achieve a margin of at least $l_{R+1} - \varepsilon$ by Corollary 3. This can only be 4ε away from ϱ^*.

We called $R + 1 = \lceil \log_2(1/\varepsilon) + 1 \rceil$ times Algorithm 2, each time calling $\lceil 2\log(N)/\varepsilon^2 + 1 \rceil$ times the base learning algorithm. Algorithm 2 returns only the last hypothesis, combining only $\lceil 2\log(N)/\varepsilon^2 + 1 \rceil$ base hypotheses. □

3.4 Infinite Hypothesis Spaces

Sofar we have implicitly assumed that the hypothesis space is finite. In this section we will show that this assumption is (often) not necessary. Also note, if the output of the hypotheses is discrete, the hypothesis space is effectively finite [10]. For *infinite hypothesis spaces*, Theorem 1 can be restated in a weaker form as:

Theorem 5 (Weak Min-Max).

$$
\gamma^* := \min_{\mathbf{d}} \sup_{h \in H} \sum_{n=1}^{N} y_n h(\mathbf{x}_n) d_n \geq \sup_{\boldsymbol{\alpha}} \min_{n=1,\ldots,N} y_n \sum_{t:\alpha_t \geq 0} \alpha_t h_t(\mathbf{x}_n) =: \varrho^*, \qquad (4)
$$

where $\mathbf{d} \in P^N$, $\boldsymbol{\alpha} \in P^{|H|}$ *with finite support. We call* $\Gamma = \gamma^* - \varrho^*$ *the "duality gap".*

In particular for any $\mathbf{d} \in P^N$: $\sup_{h \in H} \sum_{n=1}^{N} y_n h(\mathbf{x}_n) d_n \geq \gamma^*$ and for any $\boldsymbol{\alpha} \in P^{|H|}$ with finite support: $\min_{n=1,\ldots,N} y_n \sum_{t:\alpha_t \geq 0} \alpha_t h_t(\mathbf{x}_n) \leq \varrho^*$.

In theory the duality gap exists. However, Theorem 2 and Theorem 4 do not assume finite hypothesis spaces and show that the margin will converge arbitrarily close to ϱ^*, as long as the base learning algorithm can return *a single* hypothesis that has an edge not smaller than ϱ^*.

In other words, the duality gap may result from the fact that the $\sup$ on the left side can not be replaced by a $\max$, i.e. there might not exists a *single* hypothesis h with edge larger or equal to ϱ^*. By assuming that the base learner is always able to pick good enough hypotheses ($\geq \varrho^*$) one automatically gets that $\Gamma = 0$ (by Theorem 2).

Under certain conditions on H this maximum always exists and strong duality holds (see e.g. [10, 9, 6] for details):

Theorem 6 (Strong Min-Max). *If* $\{[h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)] \mid h \in H\}$ *is compact, then* $\Gamma = 0$.

In general this requirement can be fulfilled by base learning algorithms whose outputs continuously depend on the distribution $\mathbf{d}$. Furthermore, the outputs of the hypotheses need to be bounded (cf. step 3a in Algorithm 1). The first requirement might be a problem with base learning algorithms such as some variants of decision stumps or decision trees. However, there is a simple trick to avoid this problem: Roughly speaking, at each point with discontinuity $\hat{\mathbf{d}}$, one adds all hypotheses to H that are limit points of $L(S, \mathbf{d}^s)$, where $\{\mathbf{d}^s\}_{s=1}^{\infty}$ is an arbitrary sequence converging to $\hat{\mathbf{d}}$ and $L(S, \mathbf{d})$ denotes the hypothesis returned by the base learning algorithm for weighting $\mathbf{d}$ and training sample S [9].

4 Experimental Illustration

First of all, we would like to note that we are aware of the fact that maximizing the margin of the ensemble does not lead in all cases to an improved generalization performance. For fairly noisy data sets even the opposite has been reported (cf. [8, 1, 5, 11]). However, at least for well separable data the theory applies for *hard margins*. Hence, one should be able to measure differences in the generalization error, if one function approximately maximizes the margin while another function does not, as similar results in [13] on a multi-class optical character recognition problem.

Here we can report experiments on artificial data only to (a) illustrate how our algorithm works and (b) how it compares to AdaBoost. Our data is 100 dimensional and contains 98 nuisance dimensions with uniform noise. The other two dimensions are plotted exemplary in Figure 1. For training we use only 100 examples and there is obviously the need to carefully control the capacity of the ensemble.

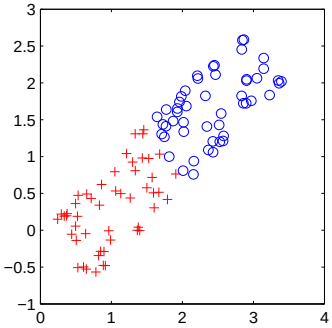

Figure 1: The two *discriminative dimensions* of our separable one hundred dimensional data set.

As base learning algorithm we use C4.5 decision trees provided by Ross Quinlan [7] using an option to control the number of nodes in the tree. We have set it such that C4.5 generates trees with about three nodes. Otherwise, the base learner often classifies all training examples correctly and over-fits the data already. Furthermore, since in this case the margin is already maximal (equal to 1), both algorithms would stop since $\hat{\gamma} = 1$. We therefore need to limit the complexity of the base learner, in good agreement with the bounds on the generalization error [13].

In Figure 2 (left) we see a typical run of Marginal AdaBoost for $\epsilon = 0.1$. It calls AdaBoost$_\varrho$ three times. The first call of AdaBoost$_\varrho$ for $\gamma = 0$ already stops after four iterations, since it has generated a consistent combined hypothesis. The lower bound l on ϱ^* as computed by our algorithm is $l = 0.07$ and the upper bound u is 0.94 (cf. step 3b in Algorithm 2). The second time ϱ is chosen to be in the middle of the interval $[l, u]$ and AdaBoost$_\varrho$ reaches the margin of $\varrho = 0.51$ after 80 iterations. The interval is now $[0.51, 0.77]$. Since the length of the interval $u - l = 0.27$ is small enough, Marginal AdaBoost leaves the loop through exit condition 4, calls AdaBoost$_\varrho$ the last time for

	C4.5	AB	Marginal AB
E_{gen}	$7.4 \pm 0.11\%$	$4.0 \pm 0.11\%$	$3.6 \pm 0.10\%$
ϱ	—	0.31 ± 0.01	0.58 ± 0.01
wins	$1/200$	$59/200$	$140/200$

Table 1 Estimated generalization performances and margins with confidence intervals for decision trees (C4.5), AdaBoost (AB) and Marginal AB on the toy data. The last row shows the number of times the algorithm had the smallest error. All numbers are averaged over 200 splits into 100 training and 19900 test examples.

$\varrho = u - \epsilon = 0.41$ and finally achieves a margin of $\hat{\varrho} = 0.55$. For comparison we also plot the margins of the hypotheses generated by AdaBoost (cf. Figure 2 (right)). One observes that it is not able to achieve a large margin efficiently ($\hat{\varrho} = 0.37$ after 1000 iterations).

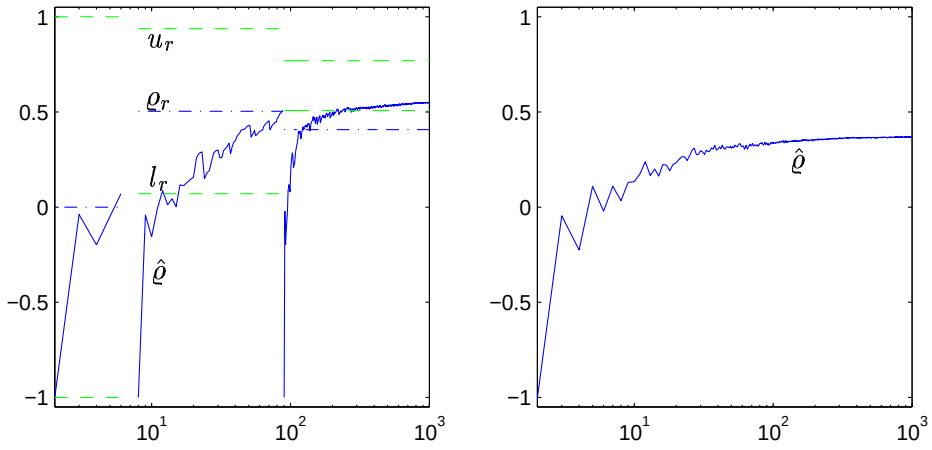

Figure 2 Illustration of the achieved margin of Marginal AdaBoost (left) and AdaBoost$_0$ (right) at each iteration. Our algorithm calls AdaBoost$_\varrho$ three times while adapting ϱ (dash-dotted). We also plot the values for l and u as in Algorithm 2 (dashed).

In Table 1 we see the average performance of the three classifiers. For AdaBoost we combined 200 hypotheses for the final prediction. For Marginal AdaBoost we use $\varepsilon = 0.1$ and let the algorithm combine only 200 hypotheses for the final prediction to get a fair comparison. We see a large improvement of both ensemble methods compared to the single classifier. There is also a slight, but – according to a T-test with confidence level 98% – significant difference between the generalization performances of both boosting algorithms. Note also that the margins of the combined hypothesis achieved by Marginal AdaBoost are on average almost twice as large as for AdaBoost.

5 Conclusion

We proposed a boosting algorithm that approximately maximizes the margin of an ensemble. To the best of our knowledge this is the first result on the *non-asymptotical convergence* of a boosting algorithm to the maximum margin solution that is valid if the hypothesis space is infinite. We have shown theoretically and empirically that our algorithm converges quite fast to the maximum margin solution, whereas the original AdaBoost algorithm is not able to achieve a large margin. We could prove this result without assuming additional properties of the base learning algorithm. In a toy experiment we have illustrated the validity of our analysis and also that a larger margin can decrease the generalization error when learning on high dimensional data with a few informative dimensions.

References

[1] L. Breiman. Prediction games and arcing algorithms. Technical Report 504, Statistics Department, University of California, December 1997.

[2] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[3] Y. Freund and R. Schapire. Game theory, on-line prediction and boosting. In *Proc. COLT*. Morgan Kaufman, 1996.

[4] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[5] A.J. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artifical Intelligence*, 1998.

[6] R. Hettich and K.O. Kortanek. Semi-infinite programming: Theory, methods and applications. *SIAM Review*, 3:380–429, September 1993.

[7] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1992.

[8] J.R. Quinlan. Boosting first-order learning. *Lecture Notes in Computer Science*, 1160:143, 1996.

[9] G. Rätsch. *Sparse ensemble learning.* PhD thesis, University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany, August 2001. in preparation.

[10] G. Rätsch, A. Demiriz, and K. Bennett. Sparse regression ensembles in infinite and finite hypothesis spaces. NeuroCOLT2 Technical Report 85, Royal Holloway College, London, September 2000. Machine Learning, to appear.

[11] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, March 2001. also NeuroCOLT Technical Report NC-TR-1998-021.

[12] R.E. Schapire. *The Desig and Analysis of Efficient Learning Algorithms.* PhD thesis, MIT Press, 1992.

[13] R.E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[14] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proc. COLT'98*, pages 80–91, 1998.

[15] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[16] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Math. Ann.*, 100:295–320, 1928.